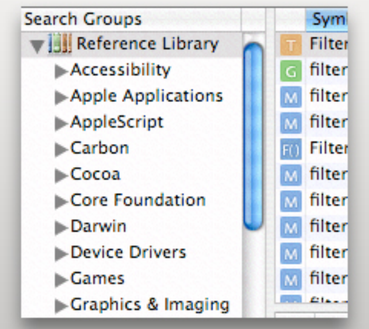
Power Programming in Xcode

Making your code work…
…and keeping it working

Session 403

These are confidential sessions—please refrain from filming or taking pictures.



Development Tools

Power Programming in Xcode

Making your code work…
…and keeping it working

Bill Bumgarner, Developer Tools Architecture
David Ewing, Developer Tools Architecture

http://developer.apple.com/wwdc2005/

Software Development Takes Too Long

- Where does the time go?

Time flies when you’re mired in tedium

Where Does the Time Go?

The Edit-Compile-Run-Debug-Repeat Loop

Software development is a highly repetitive process filled with tedious steps.
Tedium + repetition leads to silly, often time consuming, mistakes.

Using Xcode to Accelerate Development

- Reinforce the “big picture”
- Accurately suggest solutions, wherever possible
- Identify mistakes that are made
- Remember past success to reduce future re-invention

Efficient software development is all about removing much of the tedium and repetition from the process

Accelerating Development with Xcode

Diving in to the details…

- Writing code
- SCM and project configuration
- Debugging
- Testing

Writing Code

http://developer.apple.com/wwdc2005/

Writing Code

- Figure out what frameworks and APIs to use
  - And how to use them correctly!
- Know how your code works
- Create new source files and classes
- Write code accurately and quickly

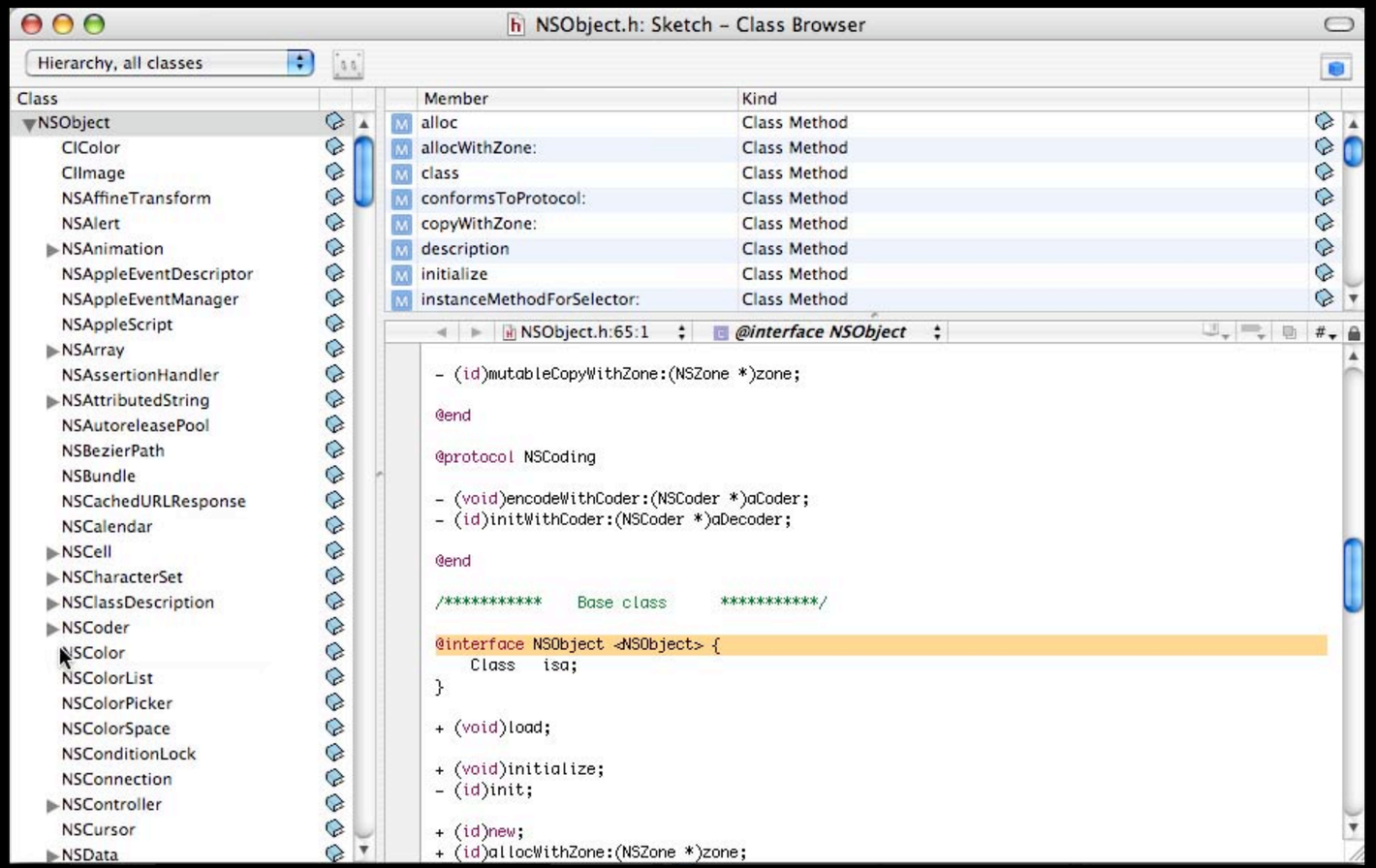
Writing Code

Figure out what frameworks and APIs to use

- Integrated Documentation Viewer Features
  - API and Full Text searches
  - Option Double-Click from source to see documentation
- Automatic updates
- Lots and lots of new content
  - Xcode user's guide
  - Man pages fully indexed
  - Many additional conceptual guides for various APIs

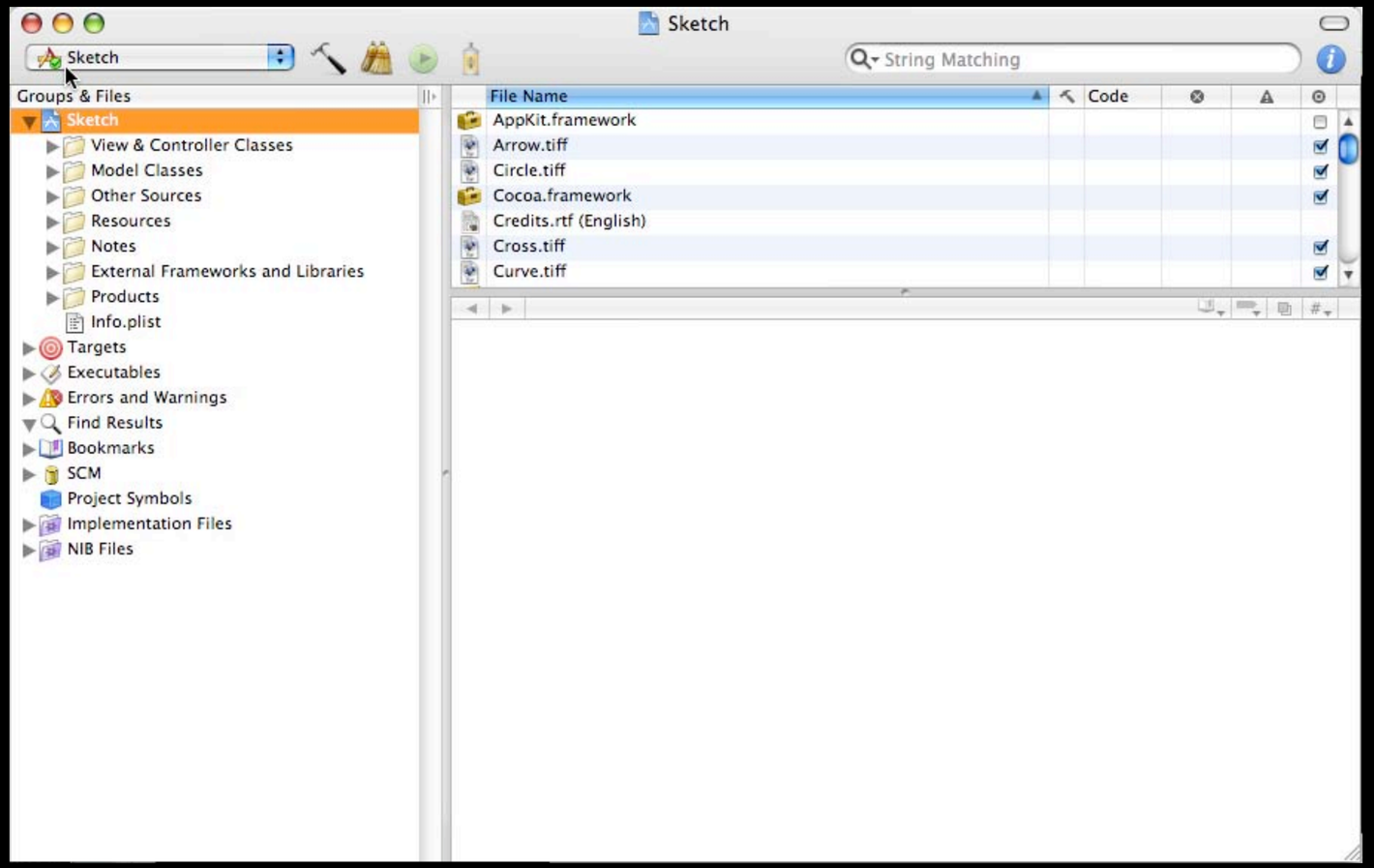
Class Browser

Looking at it all…



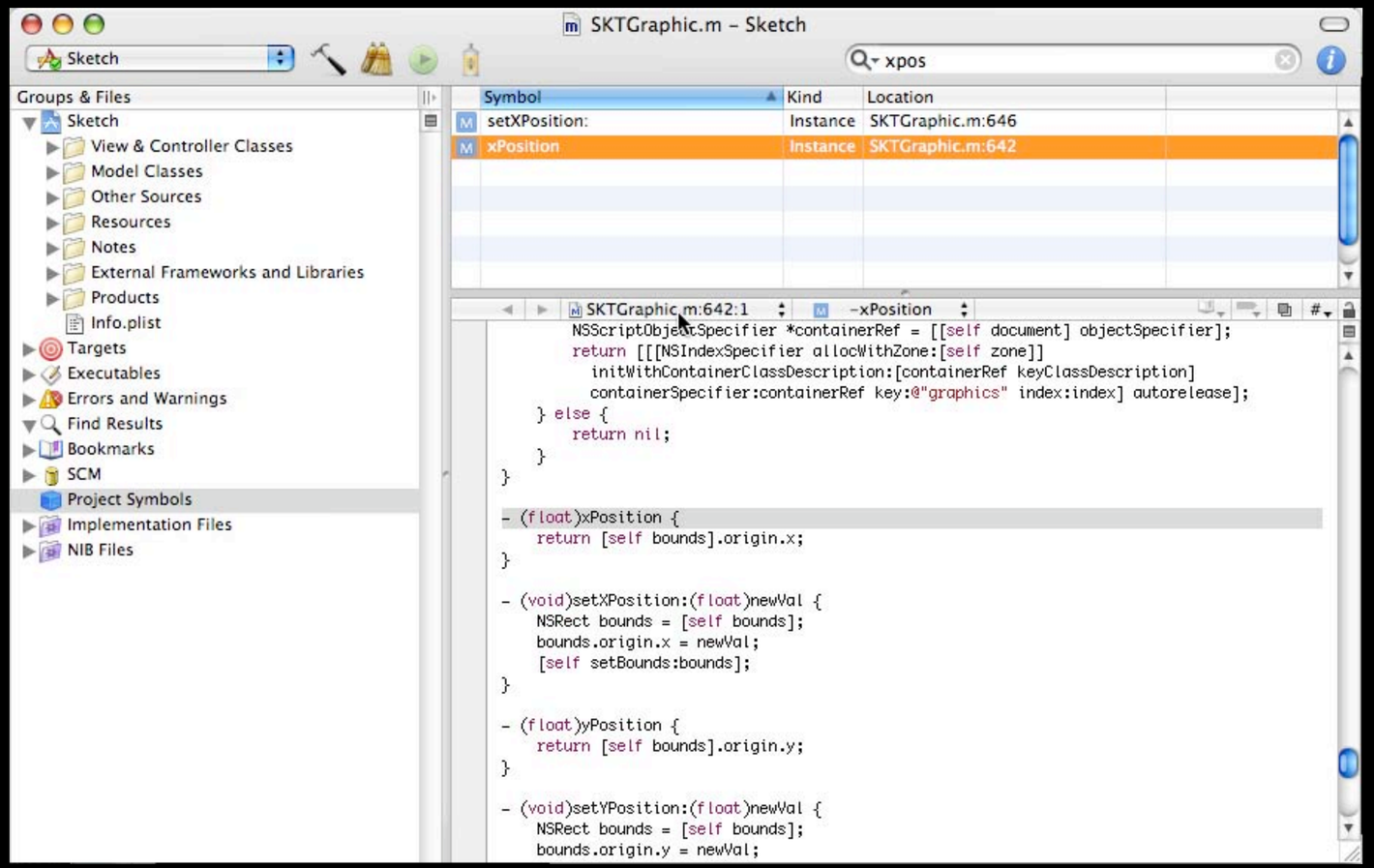
Project Symbols Smart Group

Focusing on your code…



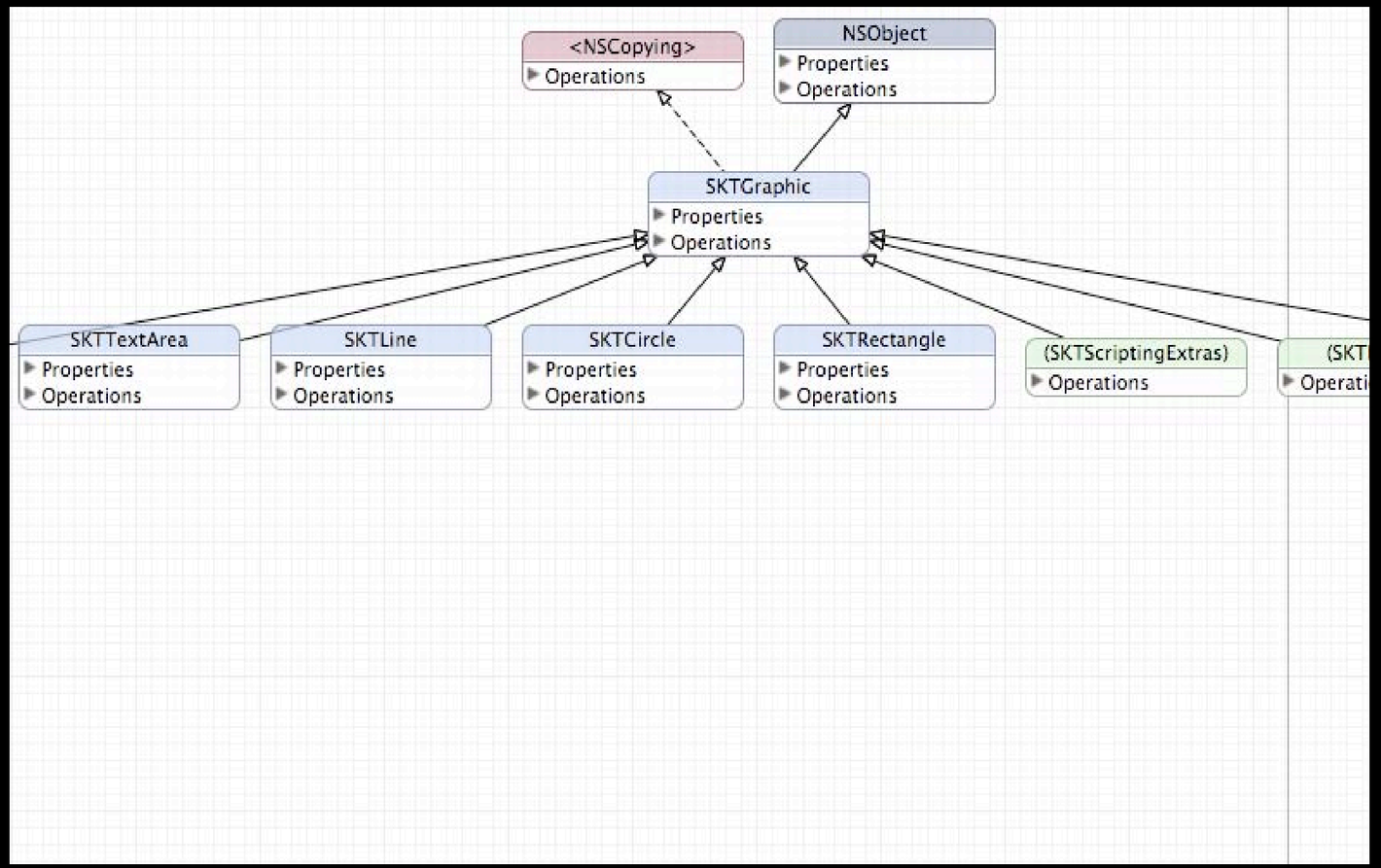
Command Double-Click

Navigating quickly…



Class Modeling

A road map to your code…



Creating New Source Files

- File templates for many types
- You can create your own
  - Examples are in "/Library/Application Support/Apple/Developer Tools/File Templates"
  - Put yours in "~/Library/Application Support/Apple/Developer Tools/File Templates"
  - Macro Expansion

```
defaults write com.apple.Xcode
PBXCustomTemplateMacroDefinitions '{ORGANIZATIONNAME = "Apple
Computer, Inc"; }'
```

Code Sense

Enter source text rapidly and accurately

- “Completion List” command
- “Next Completion” command
- Argument placeholders
- Text macros

Text Macros

Enter source text rapidly and accurately

- Available in the Edit menu
- Available via code completion
- You can add your own

Text Macros

Creating new text macros

- Defined in property list files with a .xctxtmacro extension
- Examples in “/Developer/Applications/Xcode.app/Contents/PlugIns/TextMacros.xctxtmacro”
- Put yours in “~/Library/Application Support/Apple/Developer Tools/Specifications”

Text Macros

Macros can…

- Cycle between macros for a given menu command
- Inherit settings from other macros
- Inherit settings from user defaults

```
defaults write com.apple.Xcode TextMacroDefaultSettings
   '{ c = { BlockSeparator = "\n"; }; }"'
```

Text Macros

Text macros in action…

```
- (NSArray *)graphicsWithClass:(Class)theClass {
    NSArray *graphics = [self graphics];
    NSMutableArray *result = [NSMutableArray array];



    return result;
}

- (NSArray *)rectangles {
    return [self graphicsWithClass:[SKTRectangle class]];
}

- (NSArray *)circles {
    return [self graphicsWithClass:[SKTCircle class]];
}
```

Text Macros

The gory details…

```
{
  Identifier = objc.enumerator;
  BasedOn = objc;
  IsMenuItem = YES;
  Name = "Enumerator Loop";
  TextString = "NSEnumerator * objectEnumerator = [<#collection to
    enumerate#> objectEnumerator];\n<#enumeration class#> *
    anObject;\n\nwhile ((anObject = [objectEnumerator
    nextObject]) != nil) {\n\t<#!your code here!#>\n}";
  CompletionPrefix = "NSEnum";
}
```

SCM and Project Configuration

http://developer.apple.com/wwdc2005/

Source Control Management

No change left behind

- Perforce support has been greatly improved
- Initial support for “unwrapped” files such as nib files
- Subversion support has been refined
  - Subversion, itself, has matured significantly
- Added support for working offline

Configuring Projects

- Most configuration is automatic
- Customize to speed the build process
  - Precompiled header files
  - Predictive compilation
  - Linking the executable
  - Distributed builds
- Scripting builds
  - xcodebuild command line tool
  - AppleScript support
  - Automator support

Customizing Builds

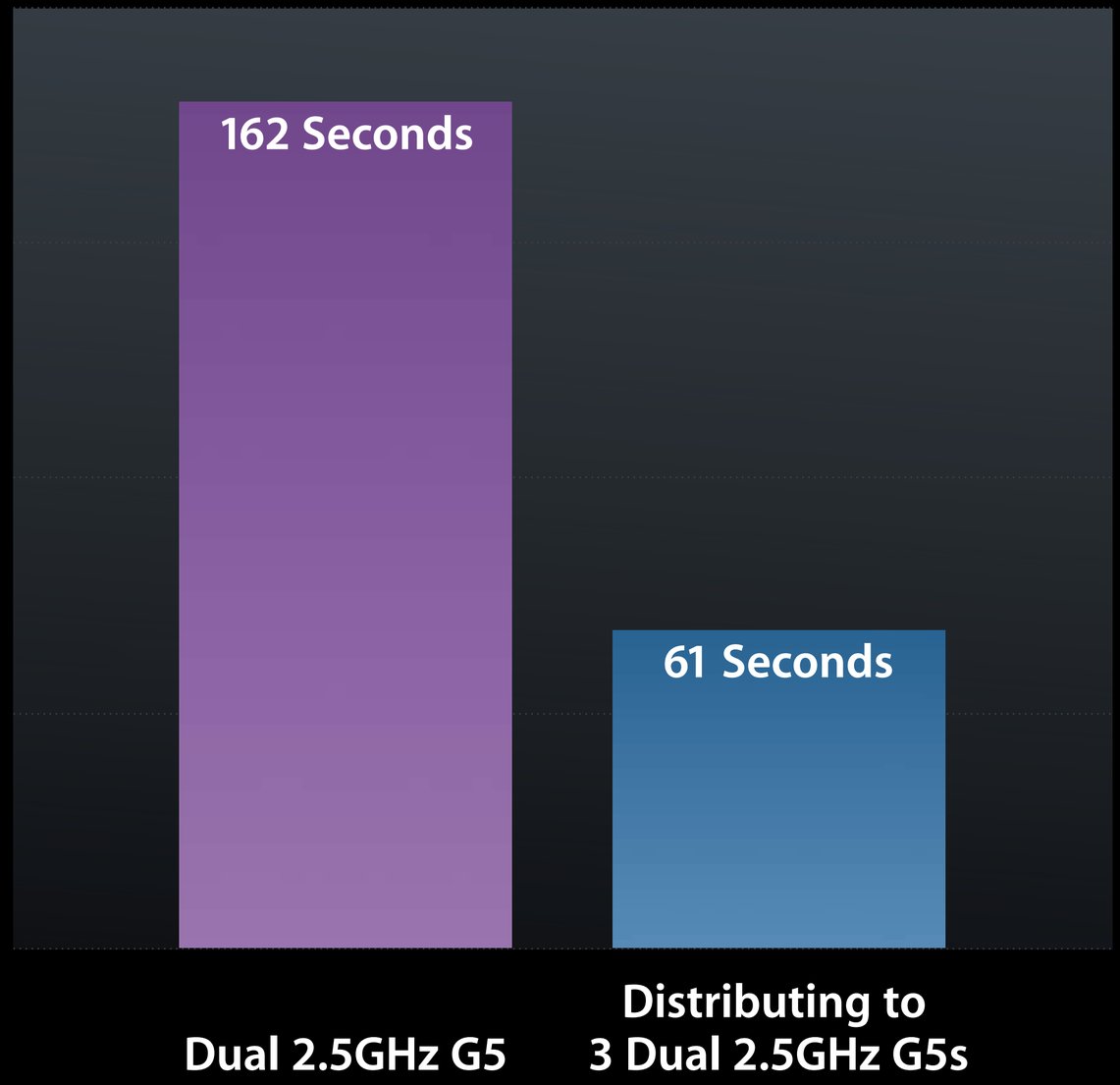
Sharing precompiled header files

- Automatic…but you can make it better
  - Prefix files must have
    - The same name
    - The same contents
    - Build settings must be the same
- Stored in /Library/Caches

Distributing Builds

Sharing the load

- Easy to set up
- Machines must be running
  - The same OS
  - The same Xcode tools
- Distributed builds are much faster
  - C++/C/ObjC Application
  - 87 source files



Scripting Xcode

- xcodebuild
  - Easy to use from shell scripts
- AppleScript (improved in 2.1)
  - Creating and configuring targets
  - Adding files
  - Running builds
  - Building multiple projects
  - Does not [yet] allow creation of projects
- Automator
  - Project builds
  - CVS commands
  - Versioning

Debugging

Debugging

- Figuring out whether it works right
- Fixing what doesn't work

It is all a matter of making the program stop when you want it to…
…and making it not stop, too!

Breakpoints

Power and control

- Breakpoint conditions
- Breakpoint actions
  - Automatically continue
    - Does not activate Xcode
- Templates and groups
- Import/Export

Breakpoint Actions

Got action?

- Debugger command
  - Sends commands to the debugger console
- Log or speak text
- Sound
  - Plays a system sound
- Shell command
  - Run a shell script
- AppleScript
  - Run an AppleScript
- Visualization breakpoint
  - Highlight method in class model

Breakpoint Conditions

Stopping (or not stopping) when you want to

- Stop if expression is true
  - Evaluates to non-zero
- Cast expressions

```
[self isKindOfClass:[SKTCircle class]]
```

```
(BOOL) [self isKindOfClass:(id)[SKTCircle class]]
```

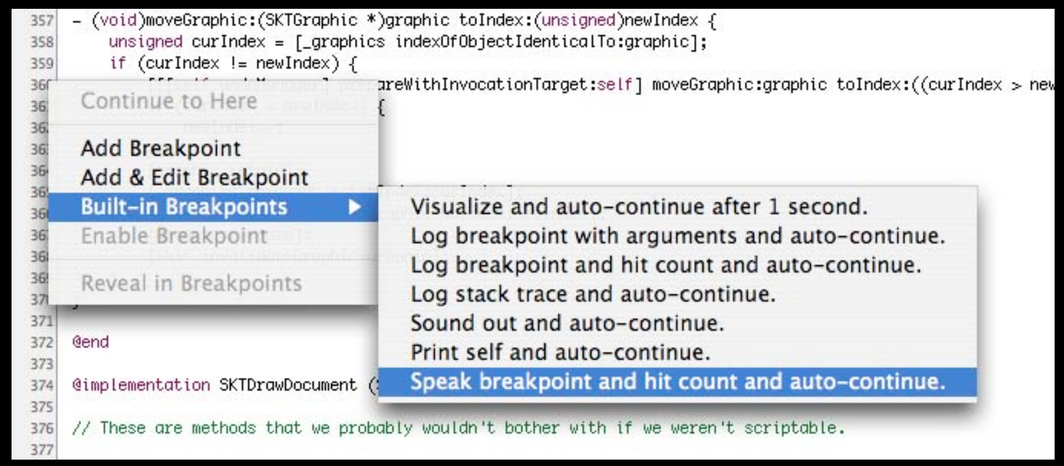
- Test in the console
- Powerful when combined with actions!

*breakpoint conditions currently only supported when debugging C/C++/ObjC/ObC++

Templates and Groups

Here come the clones

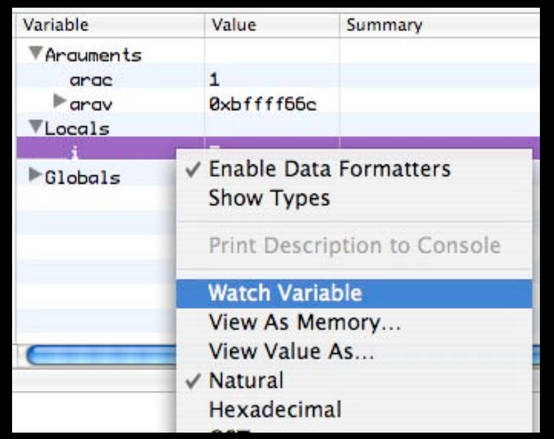
- Custom groups
- Enable/disable groups with contextual menu
- Template breakpoints
  - Control-click in gutter
- Customized templates
  - Create a ‘Template Breakpoints’ group



Watchpoints

Catch the slow wave of change

- Watch a variable via contextual menu in debugger variables view
- Marks memory page as read only
  - Slower than hardware watchpoint
  - No size limitation
- Slows execution
  - Very slow, in some cases.
- Last resort
- Dangerous to watch values on the stack
  - System calls can cause a crash
  - Watches removed routine returns



Watchpoints

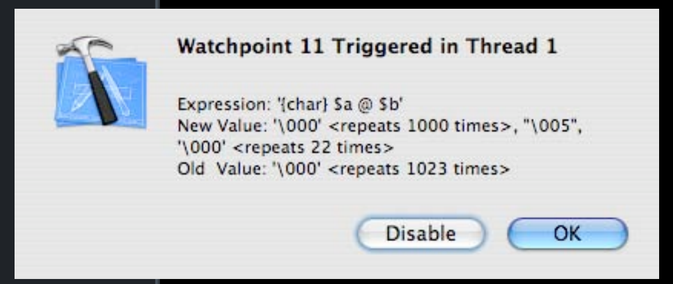
Watching a range of memory

- Given this, how do we watch for changes to the data?

```
NSMutableData *data = [NSMutableData dataWithLength: 1024];
char *bytes = [data bytes];
bytes[1000] = 5;
```

- Set a breakpoint on the second line, then in the debugger console

```
(gdb) p $a = (char *)[data bytes]
$5 = 0x1804a00 ""
(gdb) p $b = (unsigned)[data length]
$6 = 1024
(gdb) watch {char}$a@$b
Hardware watchpoint 2: {char} $a @ $b(gdb) cont
Continuing.
Watchpoint 11 Triggered in Thread 1
```



Demo

Christopher Friesen

http://developer.apple.com/wwdc2005/

Unit Testing

http://developer.apple.com/wwdc2005/

Unit Testing

- What is unit testing?
- Why write unit tests?
- Xcode Support
  - Frameworks, tools, and templates
  - Test cases and test bundles
  - Objective-C, C++, …

What Is Unit Testing?

- Small, isolated tests of individual software components
- Run fast, run regularly
- Scaffolding for your code

Why Write Unit Tests?

- Prevent bugs from creeping in and causing regressions
- Enable much more rapid change
- Act as an “executable specification”
  - Great for prototyping
  - Test-driven development

Xcode Unit Testing

- Testing frameworks and tools included
- Target templates for test bundles
- File templates for test cases
- Tests run as part of the build process
  - RunUnitTests script does the heavy lifting
- Errors are reported via build results
  - Just like compile and link errors

Xcode Unit Testing

Test cases

- Classes that contain tests implemented as methods
- Can set up and tear down a common environment (“fixture”) for the tests
  - One fixture set up and torn down around every test method
  - Fixtures separated from memory management for easier debugging
- Test methods perform assertions to verify state
- Tests are independent, don’t depend on ordering of tests
- File templates provided

Xcode Unit Testing

Test bundles

- Tests grouped in test bundles
- Clean separation between tester and testee
  - Eliminates synchronization issues of duplicated app targets
  - Easy to test both debug and release builds
- Link against the code you want to test
  - Applications—specify Bundle Loader and Test Host
- Target templates provided
  - Automatically run tests as part of a non-install build

Xcode Unit Testing

Objective-C testing

- OCUnit
  - Mature, robust framework
  - Easy to get started
- Supports applications and frameworks
- Tests run by RunUnitTests
  - Test bundle extension ".octest"
- Can test C code too

http://developer.apple.com/tools/unittest.html

Xcode Unit Testing

C++ testing

- CPlusTest
  - New framework
  - Lightweight, easy to use, standard *Unit pattern
- Supports applications and frameworks
- Tests run by RunUnitTests
  - Test bundle extension ".cptest"
- Can test C code too
- 64-bit
- Requires GCC 4.0

Xcode Unit Testing

Other testing

- Simple infrastructure for hooking in your own test tools
- “Test Rig” build setting points to custom tool
  - Test bundle path passed as parameter
  - man RunUnitTests for details on environment
- On error, generate output on stderr in GCC-like format:
  - FailingTest.c: 10: error: Caught an unexpected exception
  - WarningTest.c: 237: warning: Not run because class “Foo” is not present
- Generates error in build results

Demo

Chris Hanson

http://developer.apple.com/wwdc2005/

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

(Sub)version Control for the Rest of Us	Session 619 Wednesday 3:30PM
Power User's Guide to the Apple Developer Connection	Session 002 Thursday 9:00AM
Project Management Mastery with Xcode	Session 408 Friday 2:00PM
Development Tools Feedback Forum	Session 410 Friday 5:00PM

Q&A

Bill Bumgarner, Developer Tools Architecture
David Ewing, Developer Tools Architecture

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference